Лабораторная работа 3. Организация циклов

Задание 1. Таблица значений функции Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (см. задание 1 лабораторной работы 2), на интервале от х до х с шагом шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

Листинг кода:

using System;

namespace FunctionTable

{

class Program

{

static double CalculateFunction(double x)

{

if (x <= 0)

return x \* x;

else if (x <= 4)

return Math.Sqrt(x);

else

return 2;

}

static void Main(string[] args)

{

double xStart = -2.0;

double xEnd = 6.0;

double dx = 0.5;

Console.WriteLine("Таблица значений функции:");

Console.WriteLine("-------------------------");

Console.WriteLine("| x | f(x) |");

Console.WriteLine("-------------------------");

for (double x = xStart; x <= xEnd; x += dx)

{

double y = CalculateFunction(x);

Console.WriteLine($"| {x,7:F2} | {y,9:F2} |");

}

Console.WriteLine("-------------------------");

Console.ReadKey();

}

}

}

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Задание 2. Серия выстрелов по мишени Для десяти выстрелов, координаты которых задаются с клавиатуры, вывести текстовые сообщения о попадании в мишень из задания 2 лабораторной работы 2.

Задание 3. Ряды Тейлора Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от х до х с шагом dх с точностью в. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда

Листинг кода:

using System;

namespace TaylorSeries

{

class Program

{

static double TaylorLnRatio(double x, double eps, out int terms)

{

double sum = 0.0;

terms = 0;

if (Math.Abs(x) <= 1)

{

Console.WriteLine($"Ошибка: |x| должен быть > 1 (x = {x})");

return double.NaN;

}

int n = 0;

double term;

do

{

term = 1.0 / ((2 \* n + 1) \* Math.Pow(x, 2 \* n + 1));

sum += term;

terms++;

n++;

} while (Math.Abs(term) >= eps);

return 2 \* sum;

}

static void Main(string[] args)

{

double xStart = 1.5; // Начальное значение x (|x| > 1)

double xEnd = 3.0; // Конечное значение x

double dx = 0.2; // Шаг

double eps = 1e-6; // Точность

Console.WriteLine("Таблица значений функции ln((x+1)/(x-1)):");

Console.WriteLine("-----------------------------------------------");

Console.WriteLine("| x | f(x) | Членов ряда |");

Console.WriteLine("-----------------------------------------------");

for (double x = xStart; x <= xEnd; x += dx)

{

int terms;

double y = TaylorLnRatio(x, eps, out terms);

if (!double.IsNaN(y))

{

Console.WriteLine($"| {x,7:F2} | {y,11:F6} | {terms,11} |");

}

}

Console.WriteLine("-----------------------------------------------");

Console.ReadKey();

}

}

}

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.